

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-104993

(43)Date of publication of application : 21.04.1995

---

(51)Int.Cl. G06F 9/06

G06F 11/30

G06F 12/14

---

(21)Application number : 05-245125 (71)Applicant : MATSUSHITA ELECTRIC  
WORKS LTD

(22)Date of filing : 30.09.1993 (72)Inventor : OOKAGE SATOSHI

---

## (54) DEVELOPMENT SUPPORTING DEVICE

### (57)Abstract:

PURPOSE: To improve the reliability of a prepared real time multitask system by evading the runaway of a CPU by using an inexpensive CPU in a development supporting device for preparing the real time multitask system.

CONSTITUTION: When a writing or a reading processing is included within the program of a task by a software generation part 1 when a real time multitask system is prepared by linking each program of plural tasks, the propriety of an access is judged by referring to an access inhibition area registration table 2. Thus, if the judgment is an access impossible, the parts of the writing and reading processings are deleted from the program by a program change part 3 and error history information is extracted by an error history extraction part 4.

---

## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision  
of rejection]

[Kind of final disposal of application  
other than the examiner's decision of  
rejection or application converted  
registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's  
decision of rejection]

[Date of requesting appeal against  
examiner's decision of rejection]

[Date of extinction of right]

---

## CLAIMS

---

[Claim(s)]

[Claim 1] It is development exchange equipment for creating a real-time multitasking system which performs two or more tasks by one CPU in time sharing. The Software Generation section which creates the program of the whole system from the program of each of said task, The access keepout area registration table which registers the access keepout area for every task beforehand, The error history extract section which performs extract processing of the hysteresis information on an error when an error occurs, It comes to have the error history storage section which memorizes said extracted error history information, and the program modification section which changes the program of each of said task. The propriety of access is judged by referring to said access keepout area registration table, when it writes in in the program of said task or reading processing is included in said Software Generation section. Development exchange equipment characterized by extracting error history information in said error history extract section by said program modification section while deleting the parts of said writing and reading processing from said program when access is no.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention performs two or more tasks by one CPU in time sharing, and relates to the development exchange equipment which supports development of the real-time multitasking system corresponding to a high speed to various events.

[0002]

[Description of the Prior Art] Generally, the real-time multitasking system was a system performs two or more tasks by one CPU in time sharing, and it was made to correspond to a high speed to various events, and in order to create this real-time multitasking system, it was performing creating the program of the whole

system from the program of each task with development exchange equipment. Here, in order to improve the dependability of the created system, the overrun of a system needed to be avoided especially.

[0003] Highly efficient CPU was used conventionally and the above-mentioned trouble was solved by the memory protection feature which this CPU has. That is, in highly efficient CPU, as shown in drawing 5, each segment which constitutes real storage space has a register called a segment descriptor. In this segment descriptor, the segment base address and a limit are described and the range of a segment is limited. Here, when there is access which deviated from the range of a segment, exception interruption is generated in hard. Since each segment is specified, he is trying to set an accessible field to each task for every task.

[0004]

[Problem(s) to be Solved by the Invention] However, highly efficient CPU used as mentioned above in order to avoid an overrun had the problem of being expensive.

[0005] Making this invention in view of the above-mentioned point, the place made into the object is to offer the development exchange equipment for raising the dependability of the real-time multitasking system which avoided and created the overrun of CPU using cheap CPU.

[0006]

[Means for Solving the Problem] This invention is development exchange equipment for creating a real-time multitasking system which performs two or more tasks by one CPU in time sharing. The Software Generation section which creates the program of the whole system from the program of each of said task, The access keepout area registration table which registers the access keepout area for every task beforehand, The error history extract section which performs extract processing of the hysteresis information on an error when an error occurs, It comes to have the error history storage section which memorizes said extracted error history information, and the program modification section which changes the program of each of said task. The propriety of access is judged by referring to said access keepout area registration table, when it writes in in the program of said task or reading processing is included in said Software Generation section. When access is no, while deleting the parts of said writing and reading processing from said program, it is characterized by extracting error history information in said error history extract section by said program modification section.

[0007]

[Function] It faces linking each program of two or more tasks, and creating a real-time multitasking system, if it is in the development exchange equipment of this invention. By the Software Generation section Judge the propriety of access by referring to said access keepout area registration table, when it writes in in the program of said task or reading processing is included, and if access is improper While deleting the parts of said writing and reading processing from said program, he is trying to extract error history information in said error history extract section by said program modification section.

[0008]

[Example] Hereafter, one example of this invention is explained based on a drawing. Drawing 1 is the outline block diagram of the development exchange equipment in which one example of this invention is shown. 1 is the Software Generation section, it is faced constituting a real-time multitasking system, creates the program of the whole system from each program of two or more tasks, and completes the whole system. Since each task is the file of DOS execute form, these are linked and system-wide software is created. Furthermore, it searches whether the part which carries out the lead of data and a light is in the program of each below-mentioned task, and when there is a part which carries out a lead and a light, with reference to the below-mentioned access keepout area registration table, it detects whether this part is accessible. 2 is an access keepout area registration table, and registers the memory area which cannot access for every task. In order that it may be shown in drawing 3 and a task number may specifically identify each task, it is the number attached for every task, a field shows the address on the memory in which the program is stored, "0" shows an access failure and, as for "0" described by the lead (reading) and the light (writing) and "1", "1" shows access C. 3 is the program modification section, and it starts the below-mentioned error history extract section while it deletes the part which cannot access said program, when the signal of a purport with a part [ that it cannot access into the program for every task ] is outputted by the Software Generation section 1. 4 is the error history extract section, extracts error history information with the signal from the program modification section 3, and memorizes it in the below-mentioned error history storage section. What is necessary is for processing of NMI (non-maskable interrupt) just to perform the error history extract section 4. 5 is the error history storage section and memorizes said error history information on SRAM of memory. Here, error history information consists of information which shows the generated time, information which shows the content of each register at that time, as shown in drawing 4 .

[0009] In addition, the error history storage section 5 may be formed in the memory in a real-time multitasking system.

[0010] Next, it explains based on the flow chart which showed actuation of this example to drawing 2 . First, the access keepout area for every task is beforehand registered into the access keepout area registration table 2. And the processing program of NMI for a error history extract is stored in the error extract section 4. Here, when it searches whether the memory lead section and the memory light section are in the program of each task by the Software Generation section 1 and there is neither the memory lead section nor the memory light section, system creation processing is continued, and when there are the memory lead section and the memory light section, it investigates whether it is accessible by referring to the access keepout area registration table 2. When accessible, system creation processing is continued, and when it cannot access, while deleting the parts of a memory lead or a memory light from said program, the signal of the purport which performs error history extract processing to the error extract section 4 is outputted by the program modification section 3. In the error extract section 4,

with the signal of the purport which performs said error history extract processing, error history extract processing is performed, the extracted error history information is memorized in the error history storage section 5, and system creation is continued.

[0011] Since the part which is going to access an access keepout area is deleted from the program of each task according to the development exchange equipment of this example, by accessing an access keepout area, it is lost that a system hangs up and a reliable real-time multitasking system can be created. Furthermore, since the hysteresis information at the time of error generating is left behind, it becomes possible to analyze the cause of an overrun of a system.

[0012]

[Effect of the Invention] It faces according to the development exchange equipment of this invention, linking each program of two or more tasks, and creating a real-time multitasking system. As mentioned above, by the Software Generation section Judge the propriety of access by referring to an access keepout area registration table, when it writes in in the program of said task or reading processing is included, and if access is improper By the program modification section, while deleting the parts of said writing and reading processing from said program Since error history information was extracted in the error history extract section, the development exchange equipment for raising the dependability of the real-time multitasking system which avoided and created the overrun of CPU using cheap CPU has been offered.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] It is the outline block diagram of the development exchange equipment in which one example of this invention is shown.

[Drawing 2] It is the flow chart which shows actuation same as the above.

[Drawing 3] It is the mimetic diagram showing the keepout area registration table concerning said example.

[Drawing 4] It is the mimetic diagram showing the content of the error history storage section concerning said example.

[Drawing 5] It is the mimetic diagram showing the memory protection feature of the conventional CPU.

[Description of Notations]

- 1 Software Generation Section
- 2 Access Keepout Area Registration Table
- 3 Program Modification Section
- 4 Error History Extract Section
- 5 Error History Storage Section

---

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-104993

(43) 公開日 平成7年(1995)4月21日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F	9/06	5 4 0 G	9367-5B	
	11/30	3 0 5 H	9290-5B	
	12/14	3 1 0 H		

審査請求 未請求 請求項の数 1 O L (全 7 頁)

(21) 出願番号 特願平5-245125

(22) 出願日 平成5年(1993)9月30日

(71) 出願人 000005832

松下電工株式会社

大阪府門真市大字門真1048番地

(72) 発明者 大景 聡

大阪府門真市大字門真1048番地松下電工株式会社内

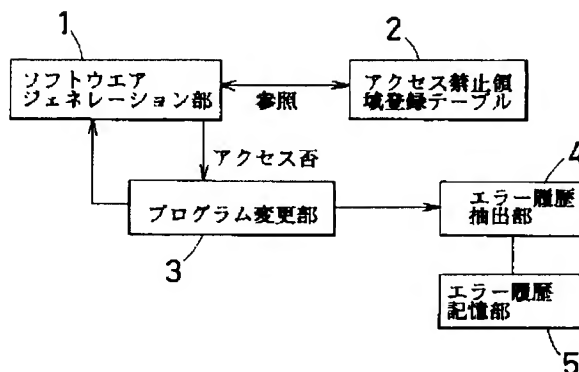
(74) 代理人 弁理士 佐藤 成示 (外 1 名)

(54) 【発明の名称】 開発支援装置

(57) 【要約】

【目的】 リアルタイムマルチタスクシステムを作成するための開発支援装置において、安価なCPUを用いて、CPUの暴走を回避し、作成したリアルタイムマルチタスクシステムの信頼性を向上させる。

【構成】 複数のタスクの各プログラムをリンクしてリアルタイムマルチタスクシステムを作成するに際し、ソフトウェアジェネレーション部1により、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合にアクセス禁止領域登録テーブル2を参照することによりアクセスの可否を判断し、アクセス不可であれば、プログラム変更部3により、前記プログラムから前記書き込み及び読み込み処理の部分を削除するとともに、エラー履歴抽出部4でエラー履歴情報を抽出するようにしている。



**【特許請求の範囲】**

【請求項 1】 1つのCPUで複数のタスクを時分割的に実行するようリアルタイムマルチタスクシステムを作成するための開発支援装置であって、前記各タスクのプログラムからシステム全体のプログラムを作成するソフトウェアジェネレーション部と、予め、タスク毎のアクセス禁止領域を登録しておくアクセス禁止領域登録テーブルと、エラーが発生したときにエラーの履歴情報の抽出処理を行うエラー履歴抽出部と、前記抽出したエラー履歴情報を記憶しておくエラー履歴記憶部と、前記各タスクのプログラムを変更するプログラム変更部とを有してなり、前記ソフトウェアジェネレーション部では、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセスが否の場合には、前記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除するとともに、前記エラー履歴抽出部でエラー履歴情報を抽出するようにしたことを特徴とする開発支援装置。

**【発明の詳細な説明】****【0001】**

【産業上の利用分野】 本発明は、1つのCPUで複数のタスクを時分割的に実行し、様々な事象に対して高速に対応するリアルタイムマルチタスクシステムの開発を支援する開発支援装置に関するものである。

**【0002】**

【従来の技術】 一般に、リアルタイムマルチタスクシステムは、1つのCPUで複数のタスクを時分割的に実行し、様々な事象に対して高速に対応するようにしたシステムであり、このリアルタイムマルチタスクシステムを作成するためには、開発支援装置により、各タスクのプログラムからシステム全体のプログラムを作成するというを行っていた。ここで、作成したシステムの信頼性を向上するために、特に、システムの暴走を回避する必要があった。

【0003】 従来は、高機能なCPUを使用し、このCPUが有しているメモリ保護機能により、上記の問題点を解決していた。つまり、高機能なCPUにおいて、図5に示すように、実メモリ空間を構成する各セグメントはセグメント・ディスクリプタというレジスタを有している。このセグメント・ディスクリプタの中には、セグメント・ベース・アドレス及びリミットが記述されており、セグメントの範囲が限定されている。ここで、セグメントの範囲から逸脱したアクセスがあった場合には、ハード的に例外割り込みを発生させるようになっている。各タスクには、それぞれのセグメントが規定されるので、タスク毎にアクセス可能領域を設定するようにしている。

**【0004】**

【発明が解決しようとする課題】 ところが、上述のように、暴走を回避するために使用する高機能なCPUは、高価であるという問題があった。

【0005】 本発明は、上記の点に鑑みてなしたものであり、その目的とするところは、安価なCPUを用いて、CPUの暴走を回避し、作成したリアルタイムマルチタスクシステムの信頼性を向上させるための開発支援装置を提供することにある。

**【0006】**

【課題を解決するための手段】 本発明は、1つのCPUで複数のタスクを時分割的に実行するようリアルタイムマルチタスクシステムを作成するための開発支援装置であって、前記各タスクのプログラムからシステム全体のプログラムを作成するソフトウェアジェネレーション部と、予め、タスク毎のアクセス禁止領域を登録しておくアクセス禁止領域登録テーブルと、エラーが発生したときにエラーの履歴情報の抽出処理を行うエラー履歴抽出部と、前記抽出したエラー履歴情報を記憶しておくエラー履歴記憶部と、前記各タスクのプログラムを変更するプログラム変更部とを有してなり、前記ソフトウェアジェネレーション部では、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセスが否の場合には、前記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除するとともに、前記エラー履歴抽出部でエラー履歴情報を抽出するようにしたことを特徴とする。

**【0007】**

【作用】 本発明の開発支援装置にあつては、複数のタスクの各プログラムをリンクしてリアルタイムマルチタスクシステムを作成するに際し、ソフトウェアジェネレーション部により、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセス不可であれば、前記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除するとともに、前記エラー履歴抽出部でエラー履歴情報を抽出するようにしている。

**【0008】**

【実施例】 以下、本発明の一実施例を図面にに基づき説明する。図1は、本発明の一実施例を示す開発支援装置の概略構成図である。1はソフトウェアジェネレーション部であり、リアルタイムマルチタスクシステムを構成するに際し、複数のタスクの各プログラムからシステム全体のプログラムを作成し、全体のシステムを作り上げるものである。各タスクはDOS実行形式のファイルになっているので、これらをリンクしてシステム全体のソフトウェアを作成する。さらに、後述の各タスクのプログラム内にデータのリード、ライトをする部分があるか否



かを検索し、リード、ライトをする部分があった場合に、この部分がアクセス可能か否かを後述のアクセス禁止領域登録テーブルを参照して検出する。2はアクセス禁止領域登録テーブルであり、タスク毎のアクセスできないメモリ領域を登録しておくものである。具体的には、図3に示すようになっており、タスク番号は各タスクを識別するためにタスク毎に付された番号であり、領域はプログラムが格納されているメモリ上のアドレスを示すものであり、リード（読み込み）、ライト（書き込み）に記述された「0」、「1」は、「0」がアクセス不可を示し、「1」がアクセス可を示している。3はプログラム変更部であり、ソフトウェアジェネレーション部1により、タスク毎のプログラム内にアクセス不可の部分があった旨の信号が出力されたとき、前記プログラムのアクセス不可部分を削除するとともに、後述のエラー履歴抽出部を起動する。4はエラー履歴抽出部であり、プログラム変更部3からの信号によりエラー履歴情報の抽出を行い、後述のエラー履歴記憶部に記憶する。エラー履歴抽出部4は、例えば、NMI（ノン・アスカブル・インタラプト）の処理により行えばよい。5はエラー履歴記憶部であり、前記エラー履歴情報をメモリのSRAM上に記憶する。ここで、エラー履歴情報は、例えば、図4に示すように、発生した日時を示す情報やそのときの各レジスタの内容を示す情報等で構成される。

【0009】なお、エラー履歴記憶部5は、リアルタイムマルチタスクシステム内のメモリに形成してもよい。

【0010】次に、本実施例の動作を図2に示したフローチャートに基づき説明する。まず、予め、タスク毎のアクセス禁止領域をアクセス禁止領域登録テーブル2に登録しておく。そして、エラー履歴抽出のためのNMIの処理プログラムをエラー抽出部4に格納しておく。ここで、ソフトウェアジェネレーション部1により、各タスクのプログラム内にメモリリード部やメモリライト部があるか否かを検索し、メモリリード部やメモリライト部がない場合は、システム作成処理を続行し、メモリリード部やメモリライト部がある場合は、アクセス可能か否かをアクセス禁止領域登録テーブル2を参照することにより調べる。アクセス可能な場合はシステム作成処理を続行し、アクセスできない場合は、プログラム変更部3により、メモリリードやメモリライトの部分を前記プログラムから削除するとともに、エラー抽出部4に対してエラー履歴抽出処理を行う旨の信号を出力する。エラ

ー抽出部4では、前記エラー履歴抽出処理を行う旨の信号により、エラー履歴抽出処理を行い、抽出されたエラー履歴情報をエラー履歴記憶部5に記憶し、システム作成を続行する。

【0011】本実施例の開発支援装置によれば、各タスクのプログラムからアクセス禁止領域にアクセスしようとする部分が削除されているので、アクセス禁止領域にアクセスすることによりシステムが暴走するということなくなり、信頼性の高いリアルタイムマルチタスクシステムが作成できる。さらに、エラー発生時の履歴情報が残されるので、システムの暴走の原因を解析することが可能となる。

【0012】

【発明の効果】以上のように、本発明の開発支援装置によれば、複数のタスクの各プログラムをリンクしてリアルタイムマルチタスクシステムを作成するに際し、ソフトウェアジェネレーション部により、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合にアクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセス不可であれば、プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除するとともに、エラー履歴抽出部でエラー履歴情報を抽出するようにしたので、安価なCPUを用いて、CPUの暴走を回避し、作成したリアルタイムマルチタスクシステムの信頼性を向上させるための開発支援装置が提供できた。

【図面の簡単な説明】

【図1】本発明の一実施例を示す開発支援装置の概略構成図である。

【図2】同上の動作を示すフローチャートである。

【図3】前記実施例に係る禁止領域登録テーブルを示す模式図である。

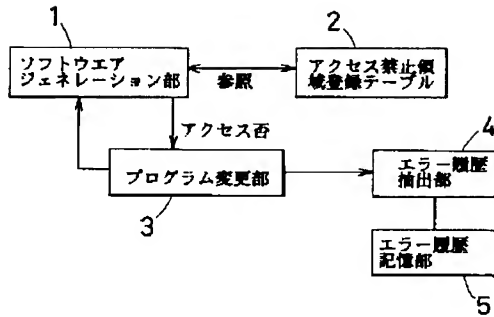
【図4】前記実施例に係るエラー履歴記憶部の内容を示す模式図である。

【図5】従来のCPUのメモリ保護機能を示す模式図である。

【符号の説明】

- 1 ソフトウェアジェネレーション部
- 2 アクセス禁止領域登録テーブル
- 3 プログラム変更部
- 4 エラー履歴抽出部
- 5 エラー履歴記憶部

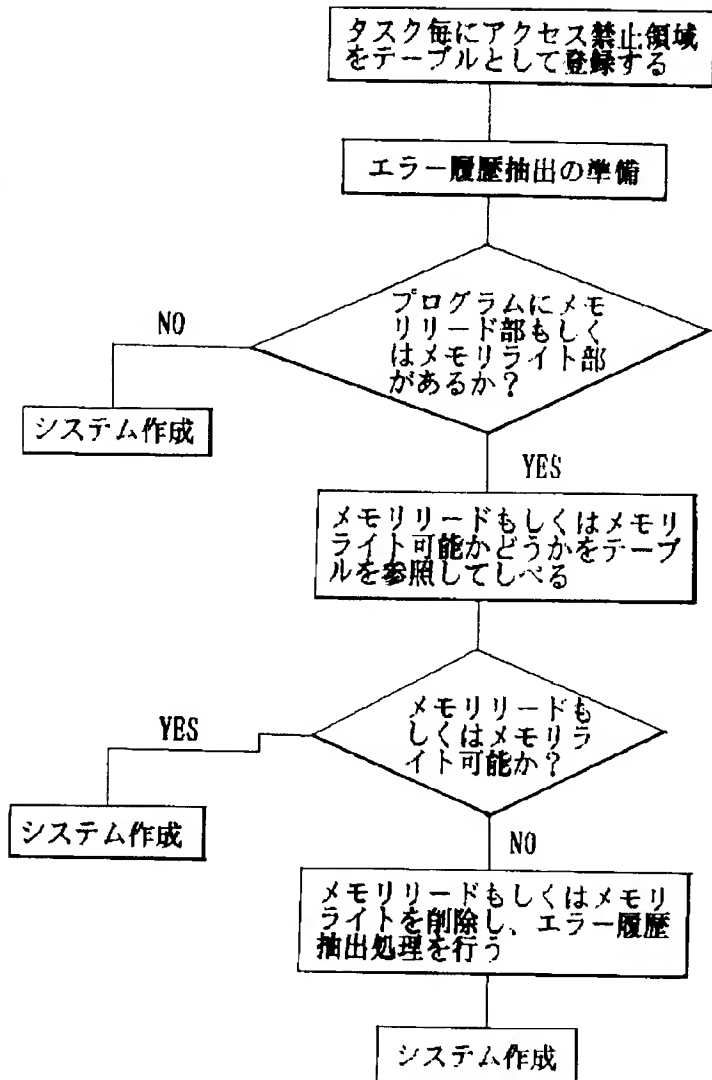
【図1】



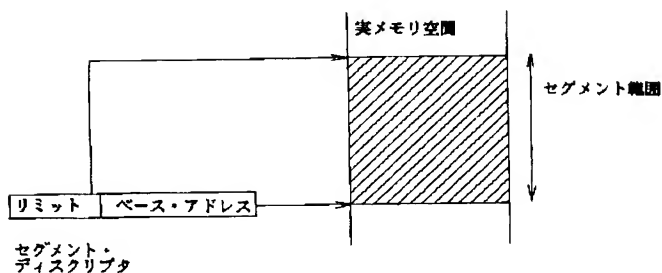
【図3】

タスク番号	領域	リード	ライト
0	0	1	0
	10000	1	0
	20000	1	0
	30000	1	0
	40000	1	0
	50000	1	0
	60000	1	0
	70000	1	1
	80000	1	1
	90000	1	1
	A0000	1	0
	B0000	1	0
	C0000	1	0
	D0000	1	0
	E0000	1	0

【図2】



【図5】



【図4】

0	最新フラグ	
2	発生回数	
4		
6	月	年
8	時	日
10	秒	分
12	予備	曜日
14	ES	
16	DS	
18	DI	
20	SI	
22	BP	
24	SP	
26	BX	
28	DX	
30	CX	
32	AX	
34	OLDIP	
36	OLDFLAG	
38	OLDCS	

## 【手続補正書】

【提出日】平成6年1月11日

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 1つのCPUで複数のタスクを時分割的に実行するようリアルタイムマルチタスクシステムを作成するための開発支援装置であって、前記各タスクのプログラムからシステム全体のプログラムを作成するソフトウェアジェネレーション部と、予め、タスク毎のアクセス禁止領域を登録しておくアクセス禁止領域登録テーブルと、エラーが発生したときにエラーの履歴情報の抽出処理を行うエラー履歴抽出部と、前記抽出したエラー履歴情報を記憶しておくエラー履歴記憶部と、前記各タスクのプログラムを変更するプログラム変更部とを有してなり、前記ソフトウェアジェネレーション部では、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセスが否の場合には、前記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分削除し、システム実行時にエラーを発生させる

処理を追加するとともに、エラーが発生した時に前記エラー履歴抽出部でエラー履歴情報を抽出するようにしたことを特徴とする開発支援装置。

## 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0006

【補正方法】変更

【補正内容】

【0006】

【課題を解決するための手段】本発明は、1つのCPUで複数のタスクを時分割的に実行するようリアルタイムマルチタスクシステムを作成するための開発支援装置であって、前記各タスクのプログラムからシステム全体のプログラムを作成するソフトウェアジェネレーション部と、予め、タスク毎のアクセス禁止領域を登録しておくアクセス禁止領域登録テーブルと、エラーが発生したときにエラーの履歴情報の抽出処理を行うエラー履歴抽出部と、前記抽出したエラー履歴情報を記憶しておくエラー履歴記憶部と、前記各タスクのプログラムを変更するプログラム変更部とを有してなり、前記ソフトウェアジェネレーション部では、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセスが否の場合には、前

記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除し、システム実行時にエラーを発生させる処理を追加するとともに、エラーが発生した時に前記エラー履歴抽出部でエラー履歴情報を抽出するようにしたことを特徴とする。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0007

【補正方法】変更

【補正内容】

【0007】

【作用】本発明の開発支援装置にあっては、複数のタスクの各プログラムをリンクしてリアルタイムマルチタスクシステムを作成するに際し、ソフトウェアジェネレーション部により、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合に前記アクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセス不可であれば、前記プログラム変更部により、前記プログラムから前記書き込み及び読み込み処理の部分を削除し、システム実行時にエラーを発生させる処理を追加するとともに、エラーが発生した時に前記エラー履歴抽出部でエラー履歴情報を抽出するようにしている。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0008

【補正方法】変更

【補正内容】

【0008】

【実施例】以下、本発明の一実施例を図面にに基づき説明する。図1は、本発明の一実施例を示す開発支援装置の概略構成図である。1はソフトウェアジェネレーション部であり、リアルタイムマルチタスクシステムを構成するに際し、複数のタスクの各プログラムからシステム全体のプログラムを作成し、全体のシステムを作り上げるものである。各タスクはDOS実行形式のファイルになっているので、これらをリンクしてシステム全体のソフトウェアを作成する。さらに、後述の各タスクのプログラム内にデータのリード、ライトをする部分があるか否かを検索し、リード、ライトをする部分があった場合に、この部分がアクセス可能か否かを後述のアクセス禁止領域登録テーブルを参照して検出する。2はアクセス禁止領域登録テーブルであり、タスク毎のアクセスできないメモリ領域を登録しておくものである。具体的には、図3に示すようになっており、タスク番号は各タスクを識別するためにタスク毎に付された番号であり、領域はプログラムが格納されているメモリ上のアドレスを示すものであり、リード（読み込み）、ライト（書き込み）に記述された「0」、「1」は、「0」がアクセス不可を示し、「1」がアクセス可を示している。3はプ

ログラム変更部であり、ソフトウェアジェネレーション部1により、タスク毎のプログラム内にアクセス不可の部分があったとき、前記プログラムのアクセス不可の部分を削除するとともに、後述のエラー履歴抽出部を起動するための信号を実行時に発生させるプログラムを追加する。4はエラー履歴抽出部であり、プログラム変更部3で変更されたプログラムを実行することによって発生する信号によりエラー履歴情報の抽出を行い、後述のエラー履歴記憶部に記憶する。エラー履歴抽出部4を起動するための信号は、例えば、NM I（ノン・マスカブル・インタラプト）で行えばよく、エラー履歴抽出部4は、システム実行時にNM Iが発生した後の割り込み処理により行えばよい。5はエラー履歴記憶部であり、前記エラー履歴情報をメモリのSRAM上に記憶する。ここで、エラー履歴情報は、例えば、図4に示すように、発生した日時を示す情報やそのときの各レジスタの内容を示す情報等で構成される。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0010

【補正方法】変更

【補正内容】

【0010】次に、本実施例の動作を図2に示したフローチャートに基づき説明する。まず、予め、タスク毎のアクセス禁止領域をアクセス禁止領域登録テーブル2に登録しておく。そして、エラー履歴抽出のためのNM Iの後処理プログラムをエラー抽出部4に格納しておく。ここで、ソフトウェアジェネレーション部1により、各タスクのプログラム内にメモリリード部やメモリライト部があるか否かを検索し、メモリリード部やメモリライト部がない場合は、システム作成処理を続行し、メモリリード部やメモリライト部がある場合は、アクセス可能か否かをアクセス禁止領域登録テーブル2を参照することにより調べる。アクセス可能な場合はシステム作成処理を続行し、アクセスできない場合は、プログラム変更部3により、メモリリードやメモリライトの部分を前記プログラムから削除し、システム実行時にエラー抽出部4に対してエラー履歴抽出処理を行う旨の信号を出力するようなプログラムを追加する。システム作成終了後、システム実行時に、エラー抽出部4では、前記エラー履歴抽出処理を行う旨の信号により、エラー履歴抽出処理を行い、抽出されたエラー履歴情報をエラー履歴記憶部5に記憶し、引き続きシステムを実行する。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0012

【補正方法】変更

【補正内容】

【0012】

【発明の効果】以上のように、本発明の開発支援装置に

よれば、複数のタスクの各プログラムをリンクしてリアルタイムマルチタスクシステムを作成するに際し、ソフトウェアジェネレーション部により、前記タスクのプログラム内に書き込みあるいは読み込み処理が含まれていた場合にアクセス禁止領域登録テーブルを参照することによりアクセスの可否を判断し、アクセス不可であれば、プログラム変更部により、前記プログラムから前記

書き込み及び読み込み処理の部分を削除し、システム実行時にエラーを発生させる処理を追加するとともに、エラーが発生した時にエラー履歴抽出部でエラー履歴情報を抽出するようにしたので、安価なＣＰＵを用いて、ＣＰＵの暴走を回避し、作成したリアルタイムマルチタスクシステムの信頼性を向上させるための開発支援装置が提供できた。